Level 1-2 Trigger Data Base development

Current status and overview

Myron Campbell, Alexei Varganov,
Stephen Miller

University of Michigan

August 17, 2000

In this presentation:

1. LOGICAL STRUCTURE OF THE DATABASE

2. OVERALL TRIGGER DB OUTLINE

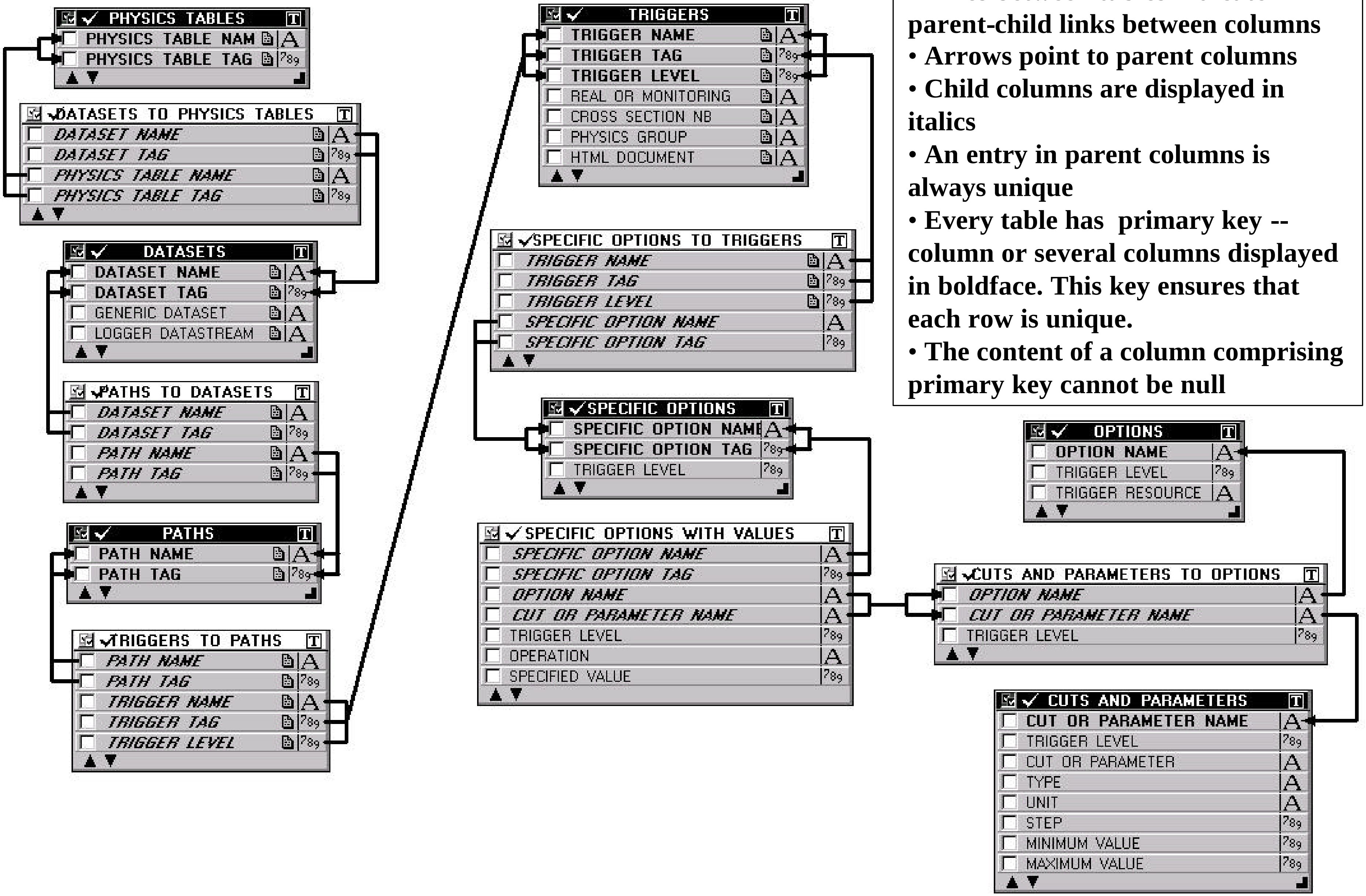
3. THE DESIGN AND “CUTTING” PROCEDURES

4. ACCESS TOOLS

5. PLANS FOR THE FUTURE

Overall Trigger Data Base outline

- Lines between tables indicate parent-child links between columns
- Arrows point to parent columns
- Child columns are displayed in italics
- An entry in parent columns is always unique
- Every table has primary key -- column or several columns displayed in boldface. This key ensures that each row is unique.
- The content of a column comprising primary key cannot be null



PHYSICS TABLES
PHYSICS TABLE NAM
PHYSICS TABLE TAG

DATASETS TO PHYSICS TABLES
DATASET NAME
DATASET TAG
PHYSICS TABLE NAME
PHYSICS TABLE TAG

DATASETS
DATASET NAME
DATASET TAG
GENERIC DATASET
LOGGER DATASTREAM

PATHS TO DATASETS
DATASET NAME
DATASET TAG
PATH NAME
PATH TAG

PATHS
PATH NAME
PATH TAG

TRIGGERS TO PATHS
PATH NAME
PATH TAG
TRIGGER NAME
TRIGGER TAG
TRIGGER LEVEL

TRIGGERS
TRIGGER NAME
TRIGGER TAG
TRIGGER LEVEL
REAL OR MONITORING
CROSS SECTION NB
PHYSICS GROUP
HTML DOCUMENT

SPECIFIC OPTIONS TO TRIGGERS
TRIGGER NAME
TRIGGER TAG
TRIGGER LEVEL
SPECIFIC OPTION NAME
SPECIFIC OPTION TAG

SPECIFIC OPTIONS
SPECIFIC OPTION NAMI
SPECIFIC OPTION TAG
TRIGGER LEVEL

SPECIFIC OPTIONS WITH VALUES
SPECIFIC OPTION NAME
SPECIFIC OPTION TAG
OPTION NAME
CUT OR PARAMETER NAME
TRIGGER LEVEL
OPERATION
SPECIFIED VALUE

OPTIONS
OPTION NAME
TRIGGER LEVEL
TRIGGER RESOURCE

CUTS AND PARAMETERS TO OPTIONS
OPTION NAME
CUT OR PARAMETER NAME
TRIGGER LEVEL

CUTS AND PARAMETERS
CUT OR PARAMETER NAME
TRIGGER LEVEL
CUT OR PARAMETER
TYPE
UNIT
STEP
MINIMUM VALUE
MAXIMUM VALUE

The design of the Data Base

•Design of the data base is performed by the Oracle Designer 2000 tool. Level 1-2 and Level 3 users are using the same design area jointly.

•The design files are set of files containing various SQL statements describing table formats, constraints, oracle procedures etc.

•Every design file is stored in offline CVS repository. Versioning of the different designs is implemented and stored in CVS.

• To set up the data base one should to run a Makefile which executes the designer files using SQLplus. This creates empty tables with the constraints and procedures. After that one should to fill up all the tables with the old content by running a special script.

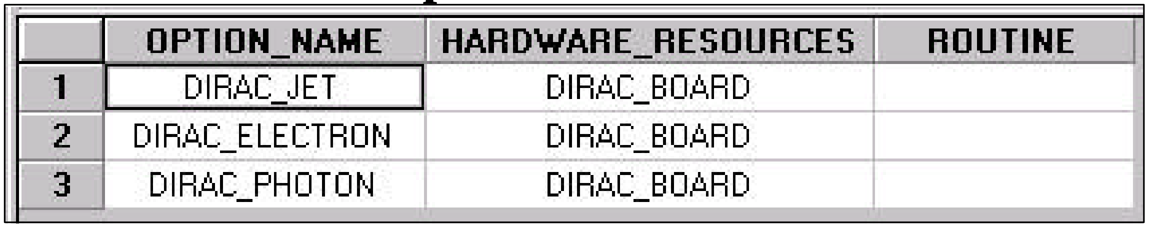
There are several Trigger Data Bases:

• DEVELOPMENT – for ongoing data base design development and access tool testing

• INTEGRATION – for merging Level 1-2 part with Level 3

• PRODUCTION – for final stable usage by RUN CONTROL and other consumers

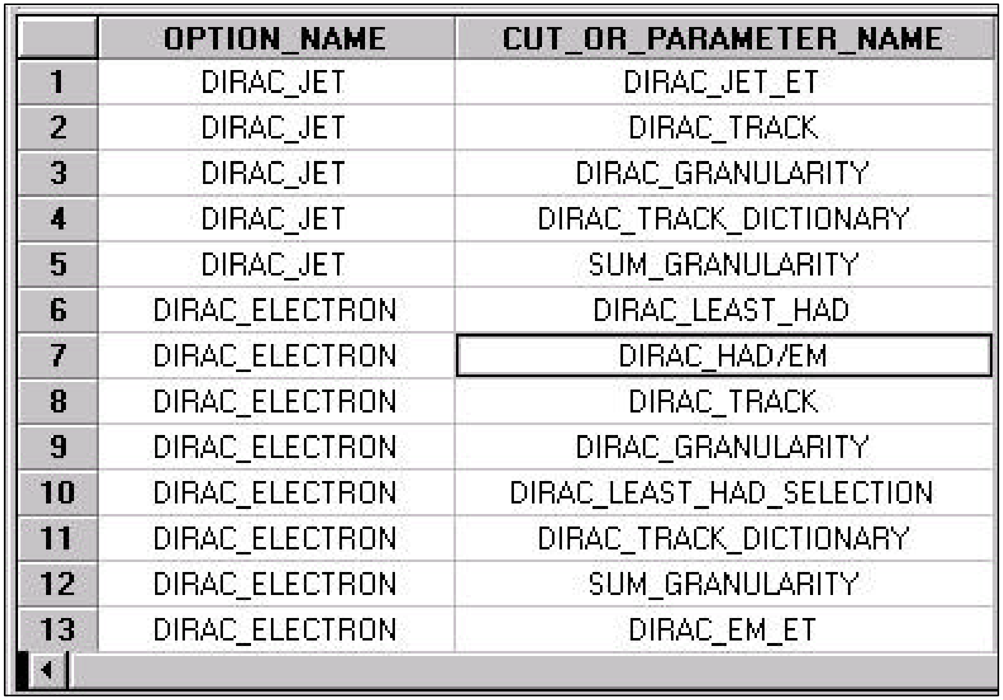
Options table



	OPTION_NAME	HARDWARE_RESOURCES	ROUTINE
1	DIRAC_JET	DIRAC_BOARD	
2	DIRAC_ELECTRON	DIRAC_BOARD	
3	DIRAC_PHOTON	DIRAC_BOARD	

- ***Options* describe physics objects and are defined by the tables OPTIONS and CUTS AND PARAMETERS TO OPTIONS**
- ***Option* is a list of *cuts* and *parameters* to be downloaded in a trigger hardware, therefore the structure of option is determined by hardware constraints**
- **One can think of the definition of an *option* as a declaration of type structure. Allocation of an instance of the declared type structure corresponds to creation of a *specific option* of the "type" option**

Cuts And Parameters to Options table



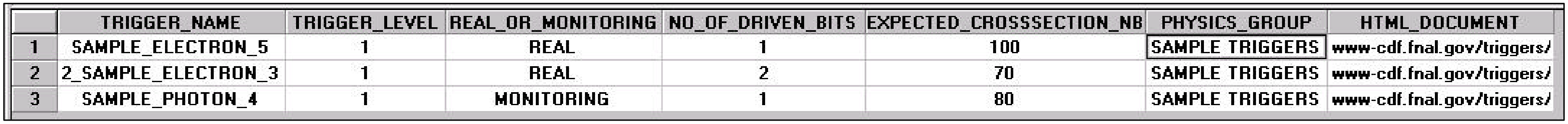
	OPTION_NAME	CUT_OR_PARAMETER_NAME
1	DIRAC_JET	DIRAC_JET_ET
2	DIRAC_JET	DIRAC_TRACK
3	DIRAC_JET	DIRAC_GRANULARITY
4	DIRAC_JET	DIRAC_TRACK_DICTIONARY
5	DIRAC_JET	SUM_GRANULARITY
6	DIRAC_ELECTRON	DIRAC_LEAST_HAD
7	DIRAC_ELECTRON	DIRAC_HAD/EM
8	DIRAC_ELECTRON	DIRAC_TRACK
9	DIRAC_ELECTRON	DIRAC_GRANULARITY
10	DIRAC_ELECTRON	DIRAC_LEAST_HAD_SELECTION
11	DIRAC_ELECTRON	DIRAC_TRACK_DICTIONARY
12	DIRAC_ELECTRON	SUM_GRANULARITY
13	DIRAC_ELECTRON	DIRAC_EM_ET

***Cuts* and *parameters* must be populated to CUTS AND PARAMETERS TO OPTIONS table according to the following rules:**

- ***Cuts* must be inserted first**
- **Then from DEPENDENCIES table *parameters* for the inserted cuts should be retrieved and inserted**
- **Finally other parameters are inserted**

The special form will be created to insert cuts and parameters to the table according to these rules

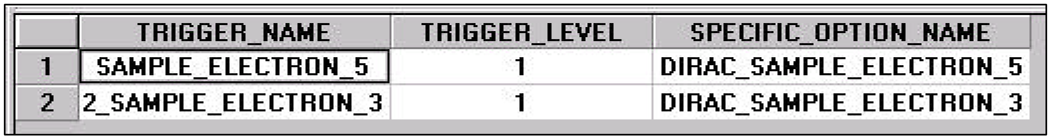
Triggers table



	TRIGGER_NAME	TRIGGER_LEVEL	REAL_OR_MONITORING	NO_OF_DRIVEN_BITS	EXPECTED_CROSSSECTION_NB	PHYSICS_GROUP	HTML_DOCUMENT
1	SAMPLE_ELECTRON_5	1	REAL	1	100	SAMPLE TRIGGERS	www-cdf.fnal.gov/triggers/
2	2_SAMPLE_ELECTRON_3	1	REAL	2	70	SAMPLE TRIGGERS	www-cdf.fnal.gov/triggers/
3	SAMPLE_PHOTON_4	1	MONITORING	1	80	SAMPLE TRIGGERS	www-cdf.fnal.gov/triggers/

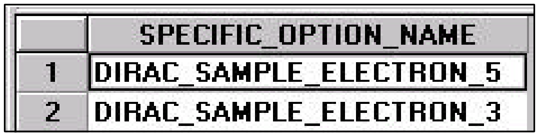
Basic definition of the *triggers*

Triggers To Specific Options table



	TRIGGER_NAME	TRIGGER_LEVEL	SPECIFIC_OPTION_NAME
1	SAMPLE_ELECTRON_5	1	DIRAC_SAMPLE_ELECTRON_5
2	2_SAMPLE_ELECTRON_3	1	DIRAC_SAMPLE_ELECTRON_3

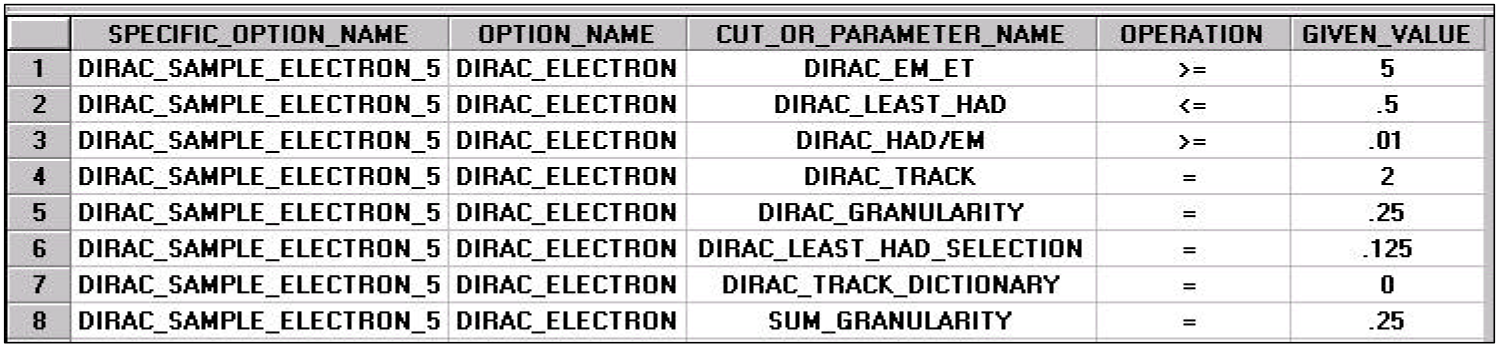
Specific Options table



	SPECIFIC_OPTION_NAME
1	DIRAC_SAMPLE_ELECTRON_5
2	DIRAC_SAMPLE_ELECTRON_3

“Allocating” of an option instance - the declaration of *specific option*

Specific Options with values table



	SPECIFIC_OPTION_NAME	OPTION_NAME	CUT_OR_PARAMETER_NAME	OPERATION	GIVEN_VALUE
1	DIRAC_SAMPLE_ELECTRON_5	DIRAC_ELECTRON	DIRAC_EM_ET	>=	5
2	DIRAC_SAMPLE_ELECTRON_5	DIRAC_ELECTRON	DIRAC_LEAST_HAD	<=	.5
3	DIRAC_SAMPLE_ELECTRON_5	DIRAC_ELECTRON	DIRAC_HAD/EM	>=	.01
4	DIRAC_SAMPLE_ELECTRON_5	DIRAC_ELECTRON	DIRAC_TRACK	=	2
5	DIRAC_SAMPLE_ELECTRON_5	DIRAC_ELECTRON	DIRAC_GRANULARITY	=	.25
6	DIRAC_SAMPLE_ELECTRON_5	DIRAC_ELECTRON	DIRAC_LEAST_HAD_SELECTION	=	.125
7	DIRAC_SAMPLE_ELECTRON_5	DIRAC_ELECTRON	DIRAC_TRACK_DICTIONARY	=	0
8	DIRAC_SAMPLE_ELECTRON_5	DIRAC_ELECTRON	SUM_GRANULARITY	=	.25

Now the *specific options* is initialized and the data structure is ready to be downloaded

The Trigger Table Java classes

The special package and product TriggerDB was created on online machine b0dau30.

The package include basic java classes for the following purposes:

- Data Base access
- Generating additional hardware related parameters for different trigger systems
- Validation of the new trigger table
- Connection to Run Control

This week GUI package will be added also to the product.

To access the product, type “setup TriggerDB”. The product directory is $TRIGGERDB_DIR

Java-generated documentation is in subdirectory /documentation

Currently the following trigger systems are included in the package and Data Base

- Global level 1 calorimeter (SUM Et, MISS Et) - calorimeter Pre FRED
- Level 1 calorimeter towers – DIRAC, calorimeter Pre FRED
- Global level 1 – FRED
- Level 2 calorimeter
- Level 1 Muon specification

Making a valid Trigger Table

To create a valid Trigger table one should:

•Specify all physics cuts and parameters such as calorimeter thresholds etc.

•Run special trigger table validation code, which checks that all the necessary cuts and parameters are fully specified and are consistent with all the "hardware rules". Once this check is complete the code generates and inserts back into the DB additional "hardware related" parameters for complete trigger specification.

• Once the trigger table is validated and fully specified it is ready to be used by Run Control code.

• Currently the validation code is only implemented for DIRAC system.

• In the perspective the similar code will be created for FRED. Right now all hardware related FRED parameters (such as bitmask) are inserted "by hands"

Plans for the future

- Complete level 1 specification by adding XTRP and XFT into the database by September
- Create calorimeter level 1 – 2 – 3 path and test the functionality of the level 1-2 part jointly with level 3. (next week?)
- Implement more complex triggers which include different hardware trigger systems.
- Finish GUI development (next week ?)
- Start working with “real” triggers rather than testing.
- Develop more trigger simulation code.